Postel

Network Working Group
Request for Comments: #202
NIC #7155
Categories: D
References: Document #2
Obsoletes: None

Steve Wolfe
UCLA-CCN
Jon Postel
UCLA-NMC
26 July 71

We have noticed a possible deadlock situation which may arise using the Initial Connection Protocol (ICP) specified in Document #2 (NIC #7101 in the Current Network Protocols Notebook NIC #7104).

If on both sides one RFC is issued and a "wait for match" is required before the second RFC is issued, it is possible that the first RFC's will not match. In particular a deadlock will occur if both sides open their send or both sides open their receive sockets first.

Briefly the ICP is:

Server	User
S1: Listen on socket L.	U1: RTS(U, L, $\underline{\ell}_1$).
S2: Wait for a match	U2: Wait for match.
S3: STR(L, U, $\underline{s}_1$)	
S4: Wait for allocation.	U3: All($\underline{\ell}_1$, $\underline{m}_1$, $\underline{b}_1$)
S5: Send data S in $\underline{s}_1$ bit bytes as allowed by allocation $\underline{m}_1$, $\underline{b}_1$.	U4: Receive data S in $\underline{s}_1$ bit bytes
S6: CLS(L, U)	U5: CLS(U, L)
S7: RTS(S, U+3, $\underline{\ell}_2$)	U6: STR(U+3, S, $\underline{s}_2$)
S8: STR(S+1, U+2, $\underline{s}_3$)	U7: RTS(U+2, S+1, $\underline{\ell}_3$)

"The labels here imply no ordering except that ordering required by the Host-Host Protocol. Note that steps S7 and S8 can be reversed as can U6 and U7. Also, notice that at any time after S2 the server could initiate steps S7 and S8 in parallel with steps S3 through S6, and that at any time after U4 the user could initiate steps U6 and U7 in parallel with step U5."

We recommend that the server perform steps 7 and 8 before waiting for the user to perform step 6 or 7. It is also suggested that the user issue the RFC's in steps 6 and 7 without waiting for the server. (If the user is only Listening then both Listens should be issued without waiting for the server.) If for some reason a host must delay between issuing RFC's it must issue the RFC's involving sockets S and U+3 first.

Network Working Group
Request for Comments: #202
NIC #7155
Categories: D
References: Document #2
Obsoletes: None

Steve Wolfe
UCLA-CCN
Jon Postel
UCLA-NMC
26 July 71

We have noticed a possible deadlock situation which may arise using the Initial Connection Protocol (ICP) specified in Document #2 (NIC #7101 in the Current Network Protocols Notebook NIC #7104).

If on both sides one RFC is issued and a "wait for match" is required before the second RFC is issued, it is possible that the first RFC's will not match. In particular a deadlock will occur if both sides open their send or both sides open their receive sockets first.

Briefly the ICP is:

Server	User
S1: Listen on socket L.	U1: RTS(U, L, ℓ_1).
S2: Wait for a match	U2: Wait for match.
S3: STR(L, U, δ_1)	
S4: Wait for allocation.	U3: All(ℓ_1, m_1, b_1)
S5: Send data S in δ_1 bit bytes as allowed by allocation m_1, b_1.	U4: Receive data S in δ_1 bit bytes
S6: CLS(L, U)	U5: CLS(U, L)
S7: RTS(S, U+3, ℓ_2)	U6: STR(U+3, S, δ_2)
S8: STR(S+1, U+2, δ_3)	U7: RTS(U+2, S+1, ℓ_3)

"The labels here imply no ordering except that ordering required by the Host-Host Protocol. Note that steps S7 and S8 can be reversed as can U6 and U7. Also, notice that at any time after S2 the server could initiate steps S7 and S8 in parallel with steps S3 through S6, and that at any time after U4 the user could initiate steps U6 and U7 in parallel with step U5."

We recommend that the server perform steps 7 and 8 before waiting for the user to perform step 6 or 7. It is also suggested that the user issue the RFC's in steps 6 and 7 without waiting for the server. (If the user is only Listening then both Listens should be issued without waiting for the server.) If for some reason a host must delay between issuing RFC's it must issue the RFC's involving sockets S and U+3 first.